

2068 PROGRAMMER'S CARTRIDGE TOOLKIT

This "toolkit" is a program which provides helpful utilities to aid in writing your own Basic programs. Because it comes on a cartridge, it offers the advantage of being instantly available (no loading from the cassette recorder). In addition, the toolkit requires only a tiny amount of RAM space so potential conflicts with other programs you use the toolkit with are kept to a minimum.

To run the toolkit, make sure the computer is turned OFF. Then insert the cartridge into the dock connector which is under the door at the front of the TS2068. Once the board is plugged in, turn the computer ON. The toolkit menu will come on the TV screen automatically. This menu lets you select one of the toolkit's 8 built-in functions. More about what these functions do and how to use them will be covered later.

In general, all functions prompt you to input the information needed to execute the procedure. Performing a toolkit function, therefore, is a simple matter of selecting the desired menu option and entering the data which the computer asks you for. In addition, the toolkit will automatically LPRINT a hard copy of each step. If you use a full size printer with one of the 6 different printer interfaces this toolkit caters for, the first thing you will want to do after turning on the computer is configure the toolkit print driver to match your brand of interface. Then, you can have hard copy of the toolkit's action printed out on your printer.

If you use the TS2040 printer, or no printer at all, it will not be necessary to configure the print driver. However, if you have a printer interface plugged into the computer, you should configure the print driver before using any of the other toolkit functions. Certain interfaces can cause the computer to hang up if this is not done. If you forget, and you find the computer just sits there without doing anything when it should be performing a function, you can always press BREAK and then enter RUN to start from the main toolkit menu again.

If you prefer not to have the automatic LPRINT feature so as not to consume paper, just turn the printer off. The toolkit will continue to LPRINT, but since the printer is not turned on, nothing gets put on paper.

1. Block Line Renumberer
2. Hex/Dec Loader
3. Disassembler
4. Tri-Base Arithmetic
5. User Defined Graphics Generator
6. Header Reader
7. Configure Print Driver
8. Back to Home Bank

DETAILS OF THE TOOLKIT FUNCTIONS

OPTION 1. Block Line Renumberer

This function will renumber a specified block of program lines in a Basic program. When you have a program in memory which needs renumbering, select option 1 from the main toolkit menu. The computer will first ask you for the "old starting line number". This represents the line in the program where renumbering is to begin. After you input the desired line number, the computer reports the following information about the line: the address of the line, its length, and its relative address which is the distance from the start of the program.

Next, the computer asks you to input the "new starting line number". This is the new value you want your block of program lines to start with. Third, the computer asks for the "step" which is the space between line numbers. Entering a step of "1" would result in program lines numbered 1, 2, 3, 4, etc. while a step of "5" would result in lines numbered 5, 10, 15, 20, 25, etc. After the step value is entered, the computer asks for the "old stopping number". This is the line number at which renumbering stops. The line with the old stopping number WILL NOT be renumbered. The renumbered block ranges from the old starting number up to but not including the old stopping number.

Once this information is input, the computer renumbers the specified lines. It prints "Renumbering complete" and takes you out of the toolkit, leaving you in the home bank with your Basic program.

When you use the Line Renumberer, two important aspects must be remembered. First, Basic lines will be renumbered, but they will not be repositioned in the listing. This means that it's possible to produce a line 1000 which comes before line 50 if you're not careful. This can result in problems when you try to edit or run the renumbered program. The best way to learn how to use the renumberer is to experiment with it by renumbering short "test" programs before you try it out on a long involved listing.

Second, although program lines are altered, RESTORE, GO TO and GO SUB commands are not. They will need to be corrected manually. To make finding these lines which contain RESTORE, GO TO, and GO SUB easy to see, the renumberer places a big black square in each line after the command. After numbering is complete, you can go through the listing and edit the lines which contain the square.

You may find that you will want to use the renumber function just to obtain the information about the program line's starting address and length without actually doing any renumbering. Other times, you may change your mind and decide not to renumber after you have selected the renumber function. Any time you wish to exit from this procedure without renumbering, just type in a zero in response to a prompt. This will take you out of the renumberer and back to the main toolkit menu.

After a block of lines have been renumbered, and the computer takes you out of the toolkit and into your Basic program, you can get back into the toolkit by typing:

POKE 23750,128:RUN or POKE 23750,128:GO TO 1

The difference between these two commands is that RUN causes the computer to CLEAR any variables your program may be storing, while GO TO starts the toolkit executing without clearing any variables. In any event, it is the POKE which is important. When memory address 23750 is poked with a 128, any RUN, GOTO, CONT, or GO SUB will go to the toolkit. When 23750 is poked with a 0, these commands will operate YOUR program in the home bank.

OPTION 2. Hex/Dec Loader

Selection 2 at the main toolkit menu turns on a handy utility which lets you read and write values to memory. You can use this for inputting those tedious machine code listings taken from magazines, or for checking your work after you have typed a listing in. The first question the computer asks you after you select option 2 is whether you intend to input Hex codes or Decimal codes. Your response should be either "H" or "D". Next the computer asks you for the address it is to start at. You can input this address either in Hex or in Decimal. This is done by prefixing the address with either a "H" or "D". For example, typing in "D6000" will signify a starting address of 6000 decimal. "H6000" will signify 6000 hex. If you forget to type in the "D" or "H", the computer will ask you to type the address again.

Once the starting address is properly input, the toolkit displays the address on the screen along with its current value. Displays are in both hex AND decimal. At the bottom of the screen, the prompt, "Hex Code?" or "Dec Code?" appears. At this point you can type in the value you wish to poke to the current address. You do not need to prefix your input with a H or D here. Just the number will do.

Several more options are possible at this input. You can press just ENTER to skip the current address and proceed to the next one. You can enter just "H" or "D" to change the numbering system for inputting values (ex: if the computer asks you for a hex code, you can type "D" to make it ask you for a decimal code instead). You can enter "A", meaning "address" to have the loader start working at a completely new address. Finally, to exit the loader, you can type "Q", or "STOP" (either the token or spelled out). Any one of these will cause the machine to go back to the main menu, but before you do, the computer asks if you wish to COPY the screen (only if you're using the TS2040 printer) and if you want to go back to the loader for some more poking. These prompts will execute only if you press "Y". If you press "N" or any other key, the program assumes you mean "NO" and takes you back to the main toolkit menu.

As your understanding of the TS2068 grows, you will find this loader utility very useful for much more than just entering machine code listings. With it, you can alter the contents of any RAM memory including Basic program lines, and variables. It is possible, for example, to use the loader to change a string array into a simple string, or a numeric array into a string array. For fun, use the Hex/Dec Loader to input random values in the Attributes file which begins at decimal address 22528. Have fun!

OPTION 3. Disassembler

The disassembler converts the values stored in memory as machine code into the more understandable mnemonic format. You can use it to find out just what's going on in the code you poke in using the hex/dec loader, or it will be helpful in unraveling the mysteries of the 2068 ROM. To use the disassembler, select option 3 at the main toolkit menu. The computer will then ask you to input the address at which you wish to start your disassembly. As in the Hex/Dec loader, precede your address with "H" or "D" depending on whether your address is a hex or a decimal number.

Once an address is entered, the program will display in hex, the addresses, the actual byte values, and the mnemonic disassembly. The disassembly will fill the screen as it progresses until the bottom line is reached. You will then be given the prompt, "more? (Y/N/C/Q)". You should enter a letter depending on what you want to do. Enter:

Y-to continue the disassembly with the next address
 N-lets you type in a NEW starting address
 C-will COPY the screen
 Q-QUITS and returns you to the main menu

OPTION 4. Tri-Base Arithmetic

The fourth option turns your computer into a programmer's multi-base arithmetic calculator capable of performing arithmetic on numbers in decimal, hexadecimal, or binary format, or even a combination of formats such as hex 2A plus binary 101001. To perform calculations, you follow the sequence of inputting a number first, an operator second (+, -, *, /, =), a second number, and another operator (such as "="). The result of the calculation will appear on the TV screen in all three number systems: hex, binary, and decimal.

To give you a better idea of the meaning of all this, here is an actual example. First press "4" and ENTER at the main toolkit menu to get into this mode. The computer will ask you for an "Input?" This can be a number preceded by a "H", "D", or "B" meaning HEX, DECIMAL, or BINARY respectively. Type in B00110101 and ENTER. The computer takes this input and converts it to 53 decimal, 0035 hex, and of course, binary 00110101. All three numbers are printed on the screen. Now type in "+" and ENTER in response to the second "Input?" prompt. The plus sign will also go up on the screen. Next enter the number "H2E6A". As before, the number is converted to binary, hex, and decimal and it's printed on the screen. Finally, press "=" and ENTER. This causes the two numbers to be summed and the result printed on the screen. At this point you could enter another "+" followed by another number to add it to this result, or you could just input a new number to start a new calculation.

The ability to perform arithmetic in any combination of hex, decimal, or binary is very useful for machine code programmers. Suppose you need to reserve space for a 312 byte table at address 40E0hex? You can let this utility calculate where the end of the table will be located. If you don't program in machine code because you can't figure out the wierd number systems involved, Tri-Base can be a very useful study tool. The binary display will give you a very illustrative demonstration of what happens to a number as the computer processes it.

The example given of two numbers being added together demonstrates only a part of the many functions available in Tri-Base. Here is a more detailed list of items you can enter at the "Input?" prompt:

Numbers-integers may be input in hex, decimal, or binary. Prefix the letters B or H to signify binary or hex. The letter D is optional for decimal inputs. If you do not precede a number with a letter, the computer assumes the number is decimal. Values must range between -65536 and +65535.

Negative numbers must be input as decimal numbers. Hex and Binary inputs must be positive, however once they are entered, they may be changed to their negative by entering the "+/-" function. Enter the three characters, "+", "/", and "-" together to change the last value displayed to its opposite. This is known as the "change sign" function on other calculators.

Arithmetic Operators-the customary "+", "-", "/", "*", and "=" functions are available. These symbols are entered alone. After entry, they appear at the far left of the screen between the two rows of values to which they apply. As noted above, the "change sign" function (+/-) is also available. This serves to give the 2's complement of a number.

Logical Operators-besides the arithmetic operators, you also have the logic operators, AND, OR, and XOR. Additionally, "C" gives the 1's complement of a number. These operators are entered the same way as the arithmetic operators.

Chaining Operations-either arithmetic or logic operations may be chained (A+B+C+D). However, there is no hierarchy among operators, and no facilities for parentheses are provided. There is a single memory store available to help with more complex calculations. This memory is accessed by entering the commands "STO" which stores the last number displayed, and "RCL" which reclaims the value and puts it on the screen.

Shift and Rotate Functions-binary numbers can be shifted left or right by entering the commands "SR" or "SL". All 16 bits of a binary number will shift and the corresponding hex and decimal values will change accordingly. Binary numbers can also have their bits rotated. Do this by entering the commands "RRn" and "RLn" where the "n" signifies the number of bits to rotate. It is necessary to include the number of bits to rotate to provide flexibility in rotating 8 bit numbers in the same way that the Z80 microprocessor does. If, for example, n=8, the left most 8 bits will be set to zero and the rotation will occur using only the right most 8 bits. This simulates the RLC x and RRC x instructions of the Z80 except that no carry flag is set or reset. If n=9, the RL x and RR x instructions can be simulated with bit 8 (the 9th bit from the right) being used as the carry flag.

Hard Copy-If your printer is connected and turned on (be sure to configure your print driver first if you're using a big printer), all displays will be Lprinted as well as displayed on the TV. Also, if you use the TS2040, you can input the command "Z" or "COPY" to copy the entire TV screen.

Returning to the main toolkit menu-is accomplished by entering the letter "Q", the token "STOP", the word "STOP" spelled out, or the letter "A" when the computer asks "Input?" All commands perform the same function; they exit Tri-Base and return you to the Main Menu.

OPTION 5. User Defined Graphics Generator

This fun to use feature lets you create user defined graphics characters with tremendous flexibility. The Graphic Generator screen shows an enlarged grid representing the 8 x 8 pixel arrangement of the first graphic character to be defined. The computer starts with the letter-A. However, you may change the character to any one you wish by typing a letter between A and U.

Below the enlarged grid you will find two identical lines displaying letters from A to U. Each letter in the top line signifies one of the definable graphic characters. The letters in the bottom line display the graphic character itself. When you first start, the two lines are the same because you haven't defined any new graphics yet. But as you create new characters, the "actual size" character will appear on the second line.

Below this is a reference telling you the effects of various keypresses. To create a new graphic character, first select the letter you wish by pressing a letter between A and U. After this character is created, you can print it on the screen by getting the "G" cursor and then typing the letter you have selected. Information about how to do this is covered on page 163 of the TS2068 owners guide.

Notice that in there is a tiny dot in the upper left square of the enlarged grid. This is a cursor which you can move about by pressing one of the arrow keys (on "5", "6", "7", and "8"). When you press just an arrow key without holding down SHIFT, the cursor will move in the direction of the arrow. When you press SHIFT and an arrow, you will also move the cursor, but you will leave a trail of INK on the grid. You can also leave a trail of PAPER by pressing the Symbol Shift key and an arrow.

The entire grid can be filled with either INK or PAPER by pressing Caps Shift and "1" (for INK) or Symbol Shift and "1" (for PAPER). You can also place a single spot of INK or PAPER at the cursor location by pressing either the "2", "3", or "9" keys simultaneously with the SHIFT or Symbol Shift keys. SHIFT will produce INK and Symbol Shift produces PAPER.

INK and PAPER can be reversed or toggled back and forth by pressing SHIFT 4 or Symbol Shift 4. Every pixel which is INK turns to PAPER and that which is PAPER becomes INK. The result is a negative or reverse image of the character being created.

To exit the Graphic Generator mode, or to clear out graphics you are not happy with, press SHIFT and "0". The computer will then ask you if you want to restore the original graphics. If you press "y" in response to this prompt, the effect is to erase every User Defined Graphic character and replace it with the original alpha character. The computer lets you start out with a clean slate. If you enter "n" in response to this prompt, the machine next asks you if you want to SAVE your graphics on tape. A "y" response here will cause the graphics to be saved so you can load them into another program. After a save, you can either go back to the main toolkit menu or create more graphics. If you do not save to tape, the program goes directly back to the main menu. Your user defined graphics will remain intact as long as you don't turn off the computer. You can go back later to work on more UDG's if you wish.

OPTION 6: Header Reader

The Header Reader function allows you to read the header data on cassette tapes. It provides information about what type of data is on the cassette like the Name, whether it is a Basic program, a character array, code, etc. Where applicable, the function tells what line the program will auto start from, the address and number of bytes of code, and the length of variables area of Basic programs. This type of information can be very useful if you are trying to break into a protected program to see how it works. The header reader's analysis can be very informative in providing details about the nature of various programs, machine code, or data you have on cassette.

When you select this option, the computer will ask you to play the tape you want to read. Once read, a display of all pertinent information is printed on the screen. If your printer is turned on, this data will also be Lprinted. If you have set the toolkit for use with the TS2040 printer, you will also be given the option to COPY the screen. After you respond to these prompts, you can either read another header or go back to the toolkit menu.

OPTION 7: Configure Print Driver

If you use a full size printer with your computer, you will definitely want to make use of the toolkit's built-in printer driver software. With it, you can LPRINT to using any of the 6 commercially available printer interfaces without having to load separate cassette based driver software. Also, because the toolkit drivers are located in the dock bank, conflicts and incompatibility problems are kept to a minimum.

To make use of the toolkit print drivers, select option 7 at the main menu. This will cause a screen to come up which offers you a choice of interfaces. Enter the number representing the brand you use. Note that the last option is listed as "Other". This gives you the ability to provide your own driver program in the home bank should you use an interface which is not provided for. You will be asked to enter the entry point address of your custom driver. This is the address in the home bank where your driver is located.

Once the brand of interface is selected, you are next asked if you want the computer to make the printer drop down a line whenever a carriage return is Lprinted. With most printers, it is possible to set a switch on the printer itself to accomodate this, but sometimes, it is more convenient to have the computer do it for you. Experiment with the "y" or "n" responses to this prompt to see how your printer reacts.

Finally, the computer asks you to input which print mode you wish to use. As in selecting the brand of interface, type the number representing the desired mode. The three available modes let you Lprint in different ways. Each is useful in varying circumstances. Depending on your application, choose the mode which best fits your circumstances. Here's a description of the function of each mode:

MODE 1. Normal:

All regular characters will be printed, and tokens like AT, PRINT, GO TO, STOP, etc. will be spelled out. Uses include virtually all that would work when using the TS2040 printer except you are not restricted to 32 columns. Examples are:

```
LPRINT "Hello, there!"
LPRINT a$
LPRINT .0125
LPRINT x
LPRINT TAB 40;"test"; TAB 50;"again"
LLIST
LLIST 1000
LPRINT CHR$ 10
```

The last example above shows one way to send CHR\$ 10, (LINE FEED) one of a group of special characters called "control codes" which cause the printer to perform some special function rather than to simply smack a letter out on paper. Two other methods of lprinting control codes are possible in this print mode. The top row of of your 2068 keyboard has 8 keys which can print graphics characters on the TV screen when you put the computer into the Graphics mode by pressing SHIFT and "9". The print driver will accept these characters, but instead of sending a character to be printed, a control code is sent in its place. You have, therefore, 16 possible codes you can send when you call up the "G" cursor and press one of the graphic keys. There are 8 lower case graphics and 8 upper case graphics. The actual code sent to the printer can be determined by looking at the following table:

Graphic on the number:	sends the code:		which is the ASCII:	
	lower/upper case		lower/upper case	
1	10	31	LF	US
2	11	30	VT	RS
3	12	29	FF	GS
4	24	08	CAN	BS
5	14	19	SO	DC3
6	15	11	SI	DC1
7	01	13	HT	CR
8	27	27	ESC	ESC

These codes can be altered. The keys are assigned the above values when you decide to configure your printer interface, but if you prefer to set up your own scheme, you can by POKING a few bytes of memory. Full details appear at the end of this section.

Print mode number 1. also lets you create "Macro Commands" consisting of up to 8 different control codes or other characters. Lprinting just one code representing the macro will send all 8 control codes to the printer. Macros are created using the User Defined Graphics capabilities of the 2068. If you study the Timex owner's manual, you'll discover that a user defined graphic character is made up of 8 different bytes. To create a macro, define a graphic character so that it includes the control codes you wish to send to the printer. Then, when you lprint this character, each of the eight bytes which comprise the UDG character will be sent to the printer.

To give you a better idea of just how you might use a macro command when you lprint, consider this situation: Suppose you want to send the codes necessary to make the printer start printing enlarged, proportional, underlined text. This would require 5 separate control codes for the Prowriter which I use. They are: 14-to SHIFT OUT (enlarged type), 27 and 80 (ESC P)-proportional spacing, and 27 and 89 (ESC U)-underline.

It is possible to define a graphics character which when lprinted, will send all 5 necessary control codes in one fell swoop. The Basic lines below create UDG's for any desired letter between A and U, and lets you input the codes you want to send. Once you have created your own custom macros, they can be saved and loaded as User Defined Graphics and used in other programs.

```

10 REM Routine for creating UDG Macro Commands
20 INPUT "Type a letter A-U";L$
30 PRINT L$
40 FOR X=0 TO 7
50 PRINT X;"---";
60 INPUT "Control Code? ";Y
70 PRINT Y
80 POKE USR L$+X,Y
90 NEXT X

```

RUN this program and select the letter "A" in response to the prompt in line 20. You'll next be asked to enter 8 control codes. These are the codes you want to execute. I would enter, for example, the numbers: 14, 27, 80, 27, 89, 0, 0, 0 to make my Prowriter print enlarged underlined proportional characters.

You could also use the toolkit to define a Macro character. First, call up the Tri-Base Arithmetic option and note the binary sequence of 1's and 0's for each of the control codes you plan to send. Then, use the UDG generator to fill in the pixels of your Macro character so that the binary 1's are represented as INK while the 0's are PAPER. If your Macro does not use all 8 bytes of the UDG character, fill out the unused bytes with 0's (PAPER).

After a macro is created, send it by getting the G-cursor and typing the letter "A" (or whatever other letter your UDG is assigned to). When this character is lprinted, the 8 separate codes you input go to the printer.

MODE 2: Up-Arrow Printing.

This method of printing is called "Up-Arrow" because it involves the use of the up pointing arrow symbol (Symbol Shift-H). In this mode, normal ASCII characters (letters, numbers, and punctuation marks) are lprinted, but tokens are NOT spelled out. This makes mode 2 pretty useless for LLISTing. The advantage is that you can use mode 2 to lprint control codes embedded in a string of characters without the need for macros, definable graphics keys, or extra program lines.

A situation could arise where you are running a program such as:

```

INPUT A$
LPRINT A$

```

and you want to use the graphic keys for something besides printing control codes. This is precisely what happens in the Pro/File 2068 program. Graphic keys are used as file editing commands, and therefore, are not available for use as control codes. Mode 2, gives you another way to place control codes right inside a string. This is done using the Up Arrow, and then immediately following this character with a two digit hex number representing the control code you wish to send. For example, to send the SHIFT OUT command embedded in a string, you could do something like this:

```
LPRINT "This is a test^OETHis is another test"
```

or

```
LET A$="This is a test^OETHis is another test"
LPRINT A$
```

The hex number which follows the "^" is OE or 14 decimal which is the SHIFT OUT control code. The Up Arrow acts as a trigger to signal to the print driver that the next two characters are digits of a hex number and are to be interpreted and sent as one code rather than two.

MODE 3: Raw ASCII

Some printers have extensive graphics character sets of their own which are accessed by sending codes greater than 128 in value. Whenever print mode 1 is used, all such codes are considered tokens which are to be expanded into words like PRINT or GO TO. In Mode 3, whatever code you lprint will go to the printer unmolested. Thus mode 3 makes it possible to make use of those great graphic characters or italics which are common on so many printers. This mode does not perform any interpreting of the code you lprint like modes 1 and 2. Therefore, mode 3 is called the "Raw ASCII" mode. Examples of ways to use this mode include:

```
LPRINT " TO TO STEP STOP " (sends the codes 204, 204, 217, and 226)
LPRINT CHR$ 200             (sends the code 200)
```

LPRINT NOTES FOR PROGRAMMERS

When you select the "Configure Print Driver" option of the Toolkit menu, the interface type, Carriage Return/Linefeed option, and print mode remain in effect until you change them by selecting this option again. All options, however, can be set from within your own basic program without the need for running the toolkit. Thus, as long as the cartridge is plugged into the computer, you can make use of the built-in driver software in your own programs. To do this, first turn on the computer and let the Toolkit self start. Select option 7 and configure the driver to your specifications. Then select option 8 of the Toolkit menu which returns you to the HOME bank. Load your own program. Any commands you have which LLIST or LPRINT will call the Toolkit print driver.

- To change the print mode from within your own program, add the command

POKE 23327,xx

where xx equals 0 for NORMAL, 1 for UP ARROW, or 2 for RAW ASCII modes.

To change the printer interface from within your own Basic, POKE 23326,yy where yy equals 0 for AERCO, 1 for Tasman type-B, 2 for Tasman type-c, 4 for A&J Centronics, 5 for Byte-Back Parallel, and 6 for OTHER. (Special note for machine code programmers: In the last case, OTHER, you must also have your own print driver resident somewhere in the HOME bank of RAM, and the starting address of your code must be poked into bytes 23321 and 23322. The toolkit print driver will call your code as a subroutine so you should end it with a RET instruction. The toolkit print driver will enter your code with the character to lprint is present in the C register.)

A complete description of all variables used by the toolkit print driver appears below:

Name	Address	Description
PCOL	23320	column number store
JPAD	23321	address for OTHER print driver software
COMD	23323	Up Arrow flag
FLBT	23324	" "
CNT_	23325	" "
VER_	23326	I/F version: 0=Aerco, 1=Tasman(b), 2=Tasman(c) 4=A&J, 5=ByteBack, 6=OTHER
MODE	23327	Set Bit 7 for CR+LF, Reset Bit 7 for CR only Specifies Print Mode: 0=Normal, 1=Up Arrow, 2=Raw Ascii In ALL modes, codes 16 and 17 are treated as TAB
GRAF	23328	Start of Graphics Key Table. Graphics on keys 1-8 (in MODE 1) are definable. Poke the addresses below with the value you wish to send:

address	for key	initial value	control code
23328	GR-8	1B	Escape
23329	GR-1	0A	Line Feed
23330	GR-2	0B	Vertical Tab
23331	GR-3	0C	Form Feed
23332	GR-4	1B	Cancel
23333	GR-5	0E	Shift Out
23334	GR-6	0F	Shift In
23335	GR-7	09	Horizontal Tab
23336	Sh GR-1	0D	Carriage Return
23337	Sh GR-2	11	Device Control 1
23338	Sh GR-3	13	Device Control 3
23339	Sh GR-4	08	Back Space
23340	Sh GR-5	1D	Group Separator
23341	Sh GR-6	1E	Record Separator
23342	Sh GR-7	1F	Unit Separator
23343	Sh GR-8	1B	Escape

STOR 23346 Temporary storage for TAB spaces

USING THE TOOLKIT WITH OTHER PROGRAMS

It is the nature of toolkit type programs that they be used in conjunction with other programs also resident in memory. This is certainly possible with this program. Because this Cartridge Toolkit runs from a cartridge in the DOCK bank of the 2068, you still have virtually the entire HOME bank free to use for other Basic or machine code programs.

The last option (Number 8) of the Toolkit menu gives you the ability to stop running the Toolkit and return you to whatever you may have present in the HOME bank. Once you make this selection, the Toolkit is turned off, and the computer does not even know that the Toolkit is present in the cartridge.

Any new Basic you execute; such as adding or editing program lines, LOADING or SAVEing, use of RUN or GO TO; will affect only the HOME bank of memory. This is what makes using the Print Drivers in the cartridge so useful. All you have to do is turn on the computer, select the interface options you desire, then quit running the toolkit by selecting option 8 of the main Toolkit menu. At this point, you can LOAD the program you wish to actually use. Once it's in memory, you can RUN it, change it, SAVE it or anything you want. The toolkit will not interfere in any way. But when you or your program try to LPRINT, the cartridge print driver will be called into action.

GOING BACK INTO THE TOOLKIT

Many times you will want to make use of some of the Toolkit options AFTER you load a program into the HOME bank. Obviously, the Basic line renumber facility is not very useful if you don't have any Basic lines present! Therefore, it is clearly necessary to have a way to turn the toolkit back ON after it has been turned OFF. Here is how you do it.

Enter the command:

```
POKE 23750,128: RUN      or ....
POKE 23750,128: GO TO 1
```

Either one will bring up the main Toolkit menu. If you use the first command which uses RUN, all variables stored in memory will be cleared before the menu comes up. If you do not want to clear away any variables, use the second command.

These commands can also be included as program lines in your HOME bank program. In this way, the Toolkit can be called up automatically as a response to some sort of condition or input. Your HOME bank program could look something like:

```
10 your program lines
20 ....
30 ....
40 PRINT "Press 'T' to start TOOLKIT"
50 INPUT X$
60 IF X$="T" THEN POKE 23750,128: GO TO 1
70 your program lines
80 ....
90 ....
```


CALLING SPECIFIC TOOLKIT OPTIONS

* DIRECTLY FROM WITHIN YOUR BASIC PROGRAM

In the above commands which turn the Toolkit ON, it is the POKE 23750,128 instruction which actually makes the computer recognize the program stored in the cartridge. The RUN or GO TO instructions simply tell the computer where to start running the Toolkit. A simple RUN or GO TO 1 starts the toolkit at the beginning. It is quite possible to turn the Toolkit on and go directly to one of the program options without displaying the main menu. All you have to change is the line number following the RUN or GO TO. A list of the necessary GO TO or RUN line numbers appears below.

After you POKE 23750,128 enter:

```
GO TO 100 or RUN 100 to start the Basic Line Renumberer
GO TO 200 or RUN 200 to start the Hex/Dec Loader
GO TO 300 or RUN 300 to start the Disassembler
GO TO 500 or RUN 500 to start the Tri-Base Arithmetic
GO TO 600 or RUN 600 to start the UDG Generator
GO TO 700 or RUN 700 to start the Header Reader
GO TO 800 or RUN 800 to start the Print Driver Configuration
```

PEEKING INTO THE CARTRIDGE

You may be thinking, after looking at the above list of GO TOs, that the Toolkit cartridge must be some kind of a Basic program itself. That is exactly what it is (with a little machine code thrown in to speed things up a bit). What's more, a provision has been added to enable you to LIST the program.

This feature is rather restrictive in nature. Once you produce a listing, there is no hope of ever returning to either the HOME bank or the Toolkit program without switching OFF the computer. It is intended only to be a means for you to see what the Toolkit program looks like.

To LIST the program, select option 8 of the Toolkit menu which turns the utility OFF. Then, enter the following command:

```
POKE 23750,128:RUN 9998
```

This causes a line in the cartridge (line 9998) to fool the computer into thinking that the Cartridge program is really in the HOME bank. The computer is able to read the memory in the Eprom, but nothing will happen when it tries to write to it.

What happens after you RUN 9998 is the computer will will LIST on the TV screen. After the screen fills, you get the familiar "Scroll?" prompt at the bottom of the screen. You can continue the normal scrolling of the program listing, you can stop scrolling, and you can move the line cursor up or down by pressing the arrow keys. But since the computer is reading memory from an eprom, you cannot print anything else on the screen. If you try, the "K" cursor will simply vanish until you press enter, at which point, it starts to flash again.

RUNNING YOUR OWN MACHINE CODE WITH THE TOOLKIT

There is a very serious bug in the TS2068 ROM which severely restricts addresses which can be used for USR calls when you have any cartridge program plugged into the computer. For this reason, machine code resident in the HOME bank may not function properly when the cartridge is plugged in. Since many of you will be using the Toolkit with Machine code added in the HOME bank, you will be happy to learn that the Toolkit includes a method which circumvents the ROM bug and allows you to call machine code located anywhere in memory. You will need to go through the Basic which calls the machine code and change all the USR instructions slightly, however. Here is what you must do:

1. Load the system variable SEED with the starting address of your machine code. This would be the number following the USR function. The easiest way to do this is to use the RANDOMIZE command: RANDOMIZE XX where "XX" is your starting address.
2. Execute the command RANDOMIZE USR 25088.

As an actual example, consider the lines below:

Change this listing to...

```
10 LET A=0
20 PRINT A: LET B$="DECEMBER"
30 RANDOMIZE USR 52900
```

this.

```
10 LET A=0
20 PRINT A: LET B$="DECEMBER"
30 RANDOMIZE 52900:RANDOMIZE USR 25088
```

When this is done, machine code at address 25088 enables the HOME bank and calls the address stored in SEED. As in normal machine language calls, the address stored in SEED. As in normal machine language calls, the USR function returns with the contents of the BC register.

TIPS AND HINTS

Dealing With Errors

A great deal of error trapping has been incorporated into the Toolkit, but nothing is 100% fool-proof. If you find that you have typed in something which causes the computer to stop with an error report like "Variable not found" or "Nonsense in Basic", there are several ways to get the program running again. First, and also the simplest, is to start over completely by switching the computer OFF and then turning it back ON again. Executing the NEW command has the same effect. The computer will completely reset itself and start running again.

You may not want to take such drastic action if you have a lengthy program loaded into the HOME bank. To do so would make it necessary to load the program back into the computer. As an alternative, enter the RUN 10 or GO TO 10 to produce the Toolkit main menu again. Or, use the list of RUNs and GO TOs given previously along with the appropriate line numbers to put you back at the start of the function you were working on when the error occurred.

Note that here, you do not need to POKE 23750,128 before you start the toolkit going. When an error stops a cartridge program, the cartridge is still turned ON even though the program has stopped. A simple RUN or GO TO 1 all it takes to get it started again.

It is only in the case where you have turned OFF the cartridge by selecting option 8 that you must POKE 23750 with the value 128 before going back into the cartridge.

A Bit About Basic Variables

If you follow the procedure for making the Toolkit LIST itself, you will notice that the program uses many Basic variables in the course of its operation. In fact, almost every letter of the alphabet is used for both string and numeric variables! In most cartridge type programs, this could pose a problem if you try to store a second program in the HOME bank which uses the same variable names. You would never know if, by running the program in the cartridge, you altered the value of a variable of the same name in the HOME bank program.

The Toolkit gets around this situation by keeping the variables used in each program completely separate. Even though the Toolkit uses a string called Q\$, you can load a program into RAM which also has a Q\$. The computer will not get them mixed up. The way in which this works is fairly simple. Whenever you start the toolkit by running or going to one of the previously listed line numbers, the program turns on the Toolkit variables (this is done by the subroutine at 9997). When the program goes back to the HOME bank in the normal way (e.g. Option 8 of the main menu, or at the completion of the Basic line renumberer), the HOME variables are restored (by line 905).

As long as you enter and exit the Toolkit "according to the directions" you won't have any problems with the computer confusing HOME and Toolkit variables. There are a few "backdoor" methods to get into or out of the program, however, and they are guaranteed to cause problems if you try them.

For your information, here are the ways NOT to enter or exit the Toolkit. The values of variables used by a program in the HOME bank may be altered as a result. If you do not have a program loaded into the HOME bank, however, there is no ill effect. Feel free to experiment:

DO NOT start the program by poking 23750,128 and going to some odd ball program line not in the previous list.

DO NOT exit the Toolkit by intentionally stopping the program with an error and then using POKE 23750,0. This is a very legitimate way to turn OFF an AROS cartridge, but in the case of the Toolkit, doing so will not restore the HOME bank variables! To properly restore the variables, you must exit the cartridge by selecting option 8 of the main Toolkit menu.